

MANUAL RAPIDO DE UNIX

1. INTRODUCCION.

Características del Sistema Operativo UNIX.

- 1.2 Entrada y Salida del Sistema.**
- 1.3 Estructura de Archivos.**
- 1.4 Directorios y Archivos estándar Importantes.**
- 1.5 Sintaxis de la Línea de Comandos.**
- 1.6 Como Cambiar de Clave de Acceso.**
- 1.7 Manual en Línea.**
- 1.8 Comandos FINGER y WHO.**

2. ESTRUCTURA DE ARCHIVOS.

- 2.1 Permisos de Acceso.**
- 2.2 Desplegar Archivos.**
- 2.3 Borrado de Archivos.**
- 2.4 Copiar Archivos.**
- 2.5 Mover Archivos.**

3. DIRECTORIOS.

- 3.1 Crear Directorios.**
- 3.2Borrar Directorios.**
- 3.3Cambiar a un Directorio.**
- 3.4Renombrar Directorio.**
- 3.5Directorios . (dot) y .. (dot dot).**
- 3.6Modos de acceso al directorio.**
 - 3.6.1Lectura reg..**
 - 3.6.2Escritura (w).**
 - 3.6.3Ejecución (x).**

APÉNDICE

- I. Referencia rápida de comandos UNIX**
- II. Referencia rápida de comandos vi**
- III. Comandos de referencia telnet**

1. INTRODUCCIÓN

Al igual que otros sistemas operativos, el sistema operativo UNIX es un conjunto de programas de utilidad y un conjunto de instrumentos que permiten al usuario conectar y utilizar esas utilidades para construir sistemas y aplicaciones.

Al conjunto de programas que componen UNIX, que se encargan de proporcionar los recursos del sistema y de coordinar todos los detalles internos de la computadora se les llama en conjunto **SISTEMA OPERATIVO o KERNEL**.

UNIX se caracteriza en ser un sistema "*MULTIUSUARIO*" porque permite que dos o mas personas utilicen la computadora al mismo tiempo.

Los Usuarios se comunican con el Kernel a través de otro programa conocido como el shell. El shell es un "*Intérprete de Línea de Comandos*" que traduce los comandos tecleados por el usuario y los convierte en instrucciones que puede entender el Kernel.

1.1 Características del Sistema Operativo UNIX.

- o **KERNEL**: Este es el componente principal del sistema operativo. Se encarga de asignar tareas y manejar el almacenamiento de datos. El usuario rara vez opera directamente con el kernel, que es la parte residente en memoria del sistema operativo.
- o **SHELL**: Esta es la utilidad que procesa las peticiones de los usuarios. Cuando alguien teclaea un comando en la terminal, el shell interpreta el comando y llama el programa deseado.

El shell puede soportar múltiples usuarios, múltiples tareas, y múltiples interfaces para sí mismo. Los dos shells más populares son el BourneShell (System V) y el Cshell (BSD Unix), debido a que usuarios diferentes pueden usar diferentes shells al mismo tiempo, entonces el sistema puede aparecer diferente para usuarios

diferentes. Existe otro shell conocido como KornShell (llamado así en honor de su diseñador), que es muy popular entre los programadores.

- o **PROGRAMAS DE UTILIDAD (UTILERIAS)**: El Sistema Operativo UNIX incluye una gran variedad de programas de utilidad que pueden ser fácilmente adaptadas para realizar tareas específicas. Estas utilerías son flexibles, adaptables, portables y modulares, y pueden ser usadas junto con filtros y redireccionamientos para hacerlos más poderosos.
- o **SISTEMA MULTIUSUARIOS**: Dependiendo del equipo disponible, un UNIX puede soportar desde uno hasta más de 100 usuarios, ejecutando cada uno de ellos un conjunto diferente de programas.
- o **SISTEMA MULTITAREAS** : UNIX permite la realización de más de una tarea a la vez. Pueden ejecutarse varias tareas en su interior, mientras se presta toda la atención al programa desplegado en la terminal.
- o **ESTRUCTURA DE ARCHIVOS** : La estructura de archivos del UNIX está pensada para facilitar el registro de una gran cantidad de archivos. Utiliza una estructura jerárquica o de árbol que permite a cada usuario poseer un directorio principal con tantos subdirectorios como desee; UNIX también permite a los usuarios compartir archivos por medio de enlaces (links), que hacen aparecer los archivos en más de un directorio de usuario.

Además, UNIX permite proteger los archivos del usuario contra el acceso por parte de otros usuarios.

- o **ENTRADA Y SALIDA INDEPENDIENTE DEL DISPOSITIVO**: Los dispositivos (como una impresora o una terminal) y los archivos en disco son considerados como archivos por UNIX.

Cuando se da una instrucción al UNIX puede indicársele que envíe el resultado a cualquiera de los diversos dispositivos o archivos. Esta desviación recibe el nombre de redireccionamiento de la salida.

En forma similar, la entrada de un programa puede redireccionarse para que venga de un archivo en disco. En el UNIX, la entrada y la salida son INDEPENDIENTES DEL DISPOSITIVO, pueden redireccionarse hacia o desde cualquier dispositivo apropiado.

- o **COMUNICACIÓN ENTRE PROCESOS:** UNIX permite el uso de conductos y filtros en la línea de comandos. Un conducto (pipe) dirige la salida de un programa para que se convierta en entrada de otro. Un filtro es un programa elaborado para procesar un flujo de datos de entrada y producir otro de datos de salida. Los conductos y filtros suelen usarse para unir utilerías y realizar alguna tarea específica.

1.3 Estructura de Archivos.

Todos los datos en UNIX son organizados en archivos. Todos los archivos se organizan en subdirectorios. Estos directorios están organizados en una estructura en forma de árbol llamado el **filesystem**.

Todo archivo posee un nombre de archivo que puede componerse por uno hasta 14 caracteres (el sistema UNIX de Berkeley permite que los nombres de archivo tengan hasta 256 caracteres). El directorio raíz (**root**) debe usar siempre el símbolo **/**, y ningún otro directorio o archivo puede usar este símbolo.

Como hijos de un mismo padre, no puede haber dos directorios o archivos dentro de un mismo directorio. Pero archivos o directorios dentro de diferentes directorios si pueden llevar el mismo nombre, como hijos de padres diferentes.

La utilidad **ls** despliega todos los nombres de archivo. A un nombre de archivo que comienza con un punto se le conoce como Nombre de Archivo Invisible, pues la utilería **ls** no lo

despliega, para observar los archivos ocultos se utiliza el comando :

ls -a

1.4 Directorios y Archivos estándar Importantes.

La estructura de archivos del sistema operativo UNIX se organiza conforme a una convención, en donde generalmente se encuentra los siguientes archivos y directorios :

/ (root) El directorio raíz, que se encuentra en todos las estructuras de archivos de UNIX.

/usr Es en donde generalmente se encuentran los directorios domicilio de los usuarios.

/bin /usr/bin Estos directorios incluyen las utilerías estándar de UNIX. Por convención, **/bin** contiene las utilerías (archivos ejecutables) usadas con más frecuencia, mientras que **/usr/bin** están las utilidades más complicadas, así como programas específicos de ciertos equipos.

/dev Todos los archivos que representan dispositivos se almacenan en este directorio.

/etc Aquí se almacenan archivos varios. El más útil es el archivo **PASSWD**, que es una lista de todos los usuarios que tienen permiso para utilizar el sistema. Además de este archivo, aquí se encuentran comandos usados por el supervisor, archivos de configuración, archivos de configuración de discos, archivos de inicialización, listas válidas de usuarios, grupos, ethernet, hosts, a donde mandar mensajes críticos, etc.

/tmp Aquí se guardan los archivos temporales que crean los programas que son ejecutados.

/lib Librerías del compilador.

1.5 Sintaxis de la Línea de Comandos.

Los usuarios teclean comandos en el prompt del shell. El **prompt** por default para el **BourneShell** es el signo de dólares (**\$**). En general el shell espera ver la siguiente sintaxis :

<comando> <opciones> <argumentos>

Para iniciar la ejecución del comando, después de la sintaxis anterior se debe presionar la tecla **<ENTER>**, hasta entonces el shell interpreta la línea de comandos.

comando: Es un comando de UNIX. Algunas veces el comando es representativo de la función, por ejemplo el comando **ls** es una contracción de la primera y tercera letras de la palabra "**list**".

opciones : También conocidas como banderas (*flags*), la forma más común de ponerlas es:

- A

donde A es la abreviación de las funciones opcionales del comando; por ejemplo el comando **ls** lista el contenido de un directorio, mientras que el comando **ls -l** da una lista Larga del directorio y **ls -C** proporciona la salida en Columnas.

argumentos : Estos pueden ser nombres de archivo, nombres de usuario, o calificadores para el comando o alguna de sus opciones.

1.6 Como Cambiar de Clave de Acceso.

El comando **passwd** cambia la clave de acceso. Al ejecutar este comando (sin parámetros ni opciones) el sistema preguntará por su clave de acceso anterior, esta pregunta no se hace si el usuario no posee clave de acceso, luego el sistema pedirá la nueva clave de acceso dos veces (para evitar errores de tecleo). Si las dos claves tecleadas son iguales, entonces el sistema cambia la clave de acceso.

La nueva clave de acceso debe cumplir con ciertos requisitos que dependen de la versión y

el tipo de UNIX que se estén corriendo, en general UNIX pide que la clave de acceso tenga:

(a) *Por lo menos 6 caracteres. Solo los primeros 8 caracteres son significativos.*

(b) *Por lo menos dos caracteres alfanuméricos y al menos un caracter numérico o especial.*

(c) *La clave de acceso debe diferir del nombre del usuario (login name).*

Las claves de acceso en algunos sistemas tienen un tiempo de vida limitado, después del cual el sistema te pide que la cambies, esto con el fin de mantener un nivel de seguridad dentro del sistema.

1.7 Manual en Línea.

La mayor fuente de ayuda en línea está en forma de documentos conocidos como las páginas de manual en línea. Para acceder al manual en línea se teclaea el siguiente comando:

man <comando>

Este mandato despliega las páginas de manual del comando **<comando>**.

1.8 Comandos finger y who.

Una vez que algún usuario se encuentra dentro de un sistema UNIX, puede saber quien más está usando el sistema con uno de los siguientes comandos:

who [opciones]

Este comando, utilizado sin opciones, lista el nombre del usuario, la terminal que está usando, y la hora en que el usuario entró.

Las diferentes opciones de este comando despliegan más información acerca de los usuarios que están actualmente dentro del sistema. Alguna información disponible mediante opciones es, por ejemplo, la cantidad de tiempo que ha pasado desde que esa terminal ha tenido actividad, el identificador de procesos (**PID**) del

proceso de "**login**", comentarios e información de salida.

Un comando que provee un poco más de información acerca de los usuarios que actualmente están utilizando el sistema es :

finger [opciones] [usuario1]

Este comando sin opciones lista el login name, nombre completo, nombre de la terminal, estatus de escritura (un asterisco antes del nombre de la terminal indica que no tiene derechos de escritura), tiempo **idle** (sin actividad), tiempo que lleva conectado, lugar físico de la sesión, y número telefónico (si se conoce), todo esto para cada usuario que está conectado al sistema.

2. ESTRUCTURA DE ARCHIVOS.

INDICE

En UNIX, todos los datos están organizados en archivos, los dispositivos periféricos como por ejemplo los discos, unidades de cinta, impresoras, y terminales, se les asignan también nombres de archivo. Los archivos de dispositivo son considerados como archivos "**especiales**", porque aunque se pueden redireccionar no se puede desplegar su contenido en una terminal.

2.1 Permisos de Acceso.

Los permisos de acceso son las protecciones que se le pueden asignar a un archivo. Estas protecciones pueden proteger los archivos de cualquier usuario de que sean leídos a escritos por gente no autorizada, incluso se pueden proteger los archivos de que sean alterados por su mismo propietario (por ejemplo para evitar que los borre accidentalmente).

Hay tres tipos de usuarios que pueden acceder a un archivo :

u (user, owner) el dueño del archivo.

g (group) el grupo al que pertenece el dueño del archivo.

• **(other)** todos los demás.

Un usuario puede intentar acceder a un archivo sencillo de tres maneras: intentando leerlo, escribir en él o ejecutarlo, estos tres accesos son :

r (read) lectura; examinar, copiar datos de un archivo.

w (write) escritura; modificar, borrar un archivo.

x (execute) ejecutar; usar el archivo como un comando.

El dueño de un archivo controla que usuarios tienen permiso de acceso y de que manera pueden hacerlo; el comando **ls -l** despliega los permisos de acceso asociados a un archivo, el primer campo de 10 caracteres describe los permisos asociados con ese archivo, un campo típico se ilustra a continuación :

rwxr-xr-x

De las nueve columnas, las tres primeras representan los derechos del *usuario*, los siguientes tres son los derechos del *grupo*, y los últimos tres son los derechos de *todos* los demás. Una letra en el campo significa que el permiso (Read, Write o eXecute) esta concedido, un guión significa que el permiso no está concedido.

En el caso de directorios, que no pueden ser ejecutados, el permiso de ejecución significa que puede realizar una búsqueda por todo el directorio y listar su contenido.

Si se entra al sistema como **SUPER-USUARIO** o **root**, se tiene acceso completo a todos los archivos, independientemente del permiso del dueño o del acceso.

Para cambiar permisos de acceso de un archivo o directorio, se usa el comando **chmod** :

chmod <permiso> <archivo1[archivoN]>

El permiso se puede expresar en dos formas :

• **con letras : [ugo] [+ -=] [rwx]**

· con números: [0-7] [0-7] [0-7]

El método para cambiar un permiso con letras es el siguiente: Las letras **u, g y o** representan el Usuario, el Grupo, y Otros (todos los demás). El signo + significa que el permiso se va a conceder, mientras que el signo - significa que el permiso se va a revocar. El signo = significa que se instalen los permisos como se muestra a continuación, algunos **ejemplos** de cambio de permisos es :

\$chmod u+x,g-w,o-w archivo1

\$chmod u=rx archivo1

El segundo método interpreta el número puesto como su equivalente en binario (nueve dígitos binarios), activando el permiso donde se haya puesto un 1 y desactivándolo donde se haya puesto un 0; por ejemplo :

\$chmod 755 archivo1

Significa que al archivo 1 se le asignaron permisos de la siguiente manera :

7 5 5

111 101 101

rwx r-x r-x

Pueden establecerse dos privilegios de acceso adicionales para archivos ejecutables :

FIJAR ID (Identificación de Usuario) y FIJAR ID DE GRUPO.

Estos privilegios de acceso permiten conceder en forma temporal los privilegios de usuario, mientras el autorizado ejecuta un archivo perteneciente al usuario propietario. Fijar el permiso de acceso al archivo con la identificación da, a cualquiera que ejecute el archivo, la identidad de usuario del dueño del archivo.

2.2 Desplegar Archivos.

El comando **cat** (catenate) despliega el contenido de un archivo. Este comando lee el archivo secuencialmente y los despliega en pantalla.

cat [opciones] [archivo1[archivoN]]

Si ningún archivo es proporcionado, o se encuentra el argumento -, cat lee de la entrada estándar.

Se pueden desplegar varios archivos en el monitor uno después de otro separando los nombres de archivo por medio de un espacio.

Si el archivo tiene más líneas que una pantalla, esta se recorrerá automáticamente hasta que se haya llegado al final del archivo, para ver el contenido de un archivo por páginas se puede utilizar el comando **| more**.

pg [opciones] [archivo1[archivoN]]

El comando **tail** despliega las últimas 10 líneas de un archivo por default. Este comando acepta la opción **-N** para que despliegue las últimas N líneas.

2.3 Borrado de Archivos.

El comando **rm** borra una o más referencias a un archivo dentro de un directorio. Si esa referencia es el último enlace a un archivo, entonces el archivo será destruido. Para remover un archivo se requiere poseer permiso de escritura sobre el directorio mismo, pero no se necesita permiso de lectura o escritura sobre el archivo mismo. La sintaxis para el comando rm es :

rm [opciones] <archivo1[archivoN]>

2.4 Copiar Archivos.

Un usuario puede copiar un archivo si tiene permiso de lectura sobre ese archivo. El comando cp puede ser usado para copiar el contenido de un archivo a otro archivo.

cp <archivo1[archivoN]> <destino>

El archivo1 y destino no pueden ser el mismo, y si destino es un archivo, entonces su contenido es destruido. Si destino es un directorio, entonces el contenido de los archivos fuente es copiado a ese directorio.

El comando **cp** es no-destructivo, porque el archivo fuente nunca es alterado.

2.5 Mover Archivos.

Un usuario puede mover un archivo sólo si tiene derechos de escritura sobre ese archivo. El comando **mv** puede ser usado para renombrar un archivo.

mv <archivo1[archivoN]> <destino>

Archivo1 y destino no pueden ser el mismo, y si destino es un archivo, entonces su contenido es destruido. Si destino es un directorio, entonces el contenido de el/los archivo(s) fuente son movidos a ese directorio.

3. DIRECTORIOS.

INDICE

Un directorio es un archivo cuyo único propósito es almacenar nombres de archivos e información relacionada. Todos los archivos, ordinarios, especiales o directorios son almacenados en directorios.

El directorio donde se encuentra un usuario al hacer su primer entrada al sistema se le conoce como el "home directory". La forma de saber en qué directorio nos encontramos actualmente es utilizando el comando **pwd** que despliega el directorio actual de trabajo.

3.1 Crear Directorios.

Para crear un directorio se utiliza el comando **mkdir** :

mkdir [opciones] <nombre del directorio>

Si no se utiliza la opción para cambiar permisos de acceso, entonces el directorio tendrá los permisos de acceso por default, que son lectura, escritura y ejecución para el usuario, y lectura y ejecución para el grupo y todos los demás. Los archivos **.** y **..** son creados automáticamente. Para poder crear un sub-directorio, debes tener permiso de escritura sobre el directorio padre.

3.2 Borrar Directorios.

Para borrar un directorio se utiliza el comando **rmdir** :

rmdir [opciones] <nombre del directorio>

Normalmente, los directorios se borrarán usando el comando **rmdir**.

Antes de que el directorio pueda ser borrado, este debe estar vacío, esto es, que no deberá contener ningún archivo.

Para el propósito de borrar un directorio, el directorio esta vacío si este contiene solo dos archivos, llamados **.** (**dot**) y **..** (**dot dot**).

3.3 Cambiar a un Directorio.

Para cambiarte a un directorio dentro del sistema, usa el comando **cd** (cambio directorio).

cd <nombre del directorio>

Cuando un mensaje de error aparezca ("**cannot access file..**"), comúnmente indica que el pathname es incorrecto.

3.4 Renombrar Directorio.

Para renombrar un directorio se puede usar el comando **mv** (move).

mv <nombre directorio actual> <nuevo nombre directorio>

Los permisos en el directorio serán los mismos.

3.5 Directorios **.** (**dot**) y **..** (**dot dot**).

El archivo **.** (**dot**) representa el directorio común de trabajo; y el archivo **..** (**dot dot**) representa un directorio arriba del directorio común de trabajo, comúnmente referido como el directorio padre. Si tecleamos el comando **ls** para mostrar una lista de directorios y archivos del directorio de trabajo y usamos la opción **-a** para mostrar una lista de todos los archivos y la opción **-l** mostrará un listado largo.

El comando **ls -la** despliega los modos de acceso, número de links, el dueño, el grupo, tamaño, etc. de los archivos en un directorio;

pero también despliega las características del directorio común de trabajo y su directorio padre.

En lugar de preguntar por información en todos los archivos en un directorio, tu puedes pedir únicamente información en el directorio común de trabajo. Para esto podrás usar el **comando ls -ld**.

3.6 Modos de acceso al directorio.

Los modos de acceso están listado y organizados en la misma manera de otro archivo. Hay algunas diferencias que se van a mencionar.

3.6.1 Lectura reg..

Esto significa que al acceder a un directorio el usuario puede contener el contenido. El usuario puede ver todos lo archivos dentro del directorio.

3.6.2 Escritura (w).

Esto significa que el usuario puede crear o borrar archivos dentro de ese directorio.

3.6.3 Ejecución (x).

Esto significa que el usuario podrá ejecutar los comandos, que se encuentran dentro de dicho directorio, y hacer referencia a estos.

APÉNDICE

INDICE

I. REFERENCIA RÁPIDA DE COMANDOS DE UNIX

at -l lista los trabajos at esperando para ser ejecutados.

at -r xx elimina el trabajo at xx.

awk '/str1/,/str2/' archivo despliega todas las líneas que se encuentran contenidas entre str1 y str2 dentro del archivo.

awk '{print \$n,\$m}' archivo despliega los campos n y m de archivo.

cat archivo despliega el contenido de archivo en pantalla.

cat archivo1 >> archivo2 agrega el contenido de archivo1 al final de archivo2.

cd regresa a tu directorio home.

cd dir cambiar el directorio dir.

chmod perms archivo cambia permisos.

cp archivo1 archivo2 copia archivo1 a archivo2.

cp f1 f2 f3 dir copia los archivos f1, f2, f3 al directorio dir.

csh el C shell.

cut -fx archivo despliega el campo x de archivo.

cut -da -fx archivo usa la letra a como separador de campos.

diff archivo1 archivo2 despliega las diferencias entre archivo1 y archivo2.

echo string despliega string en la terminal.

file archivo1 describe el tipo de archivo1 (datos, texto, binario, etc..)

finger usuario despliega la información de usuario.

ftp protocolo para la transferencia de archivos.

grep string archivo busca string en archivo.

grep -c string archivo despliega únicamente el número de ocurrencias de string en el archivo.

grep -l string archivos lista los nombres de archivo que contienen string.

kill %x mata el trabajo background x.

lp -ddest archivo imprime el archivo en la impresora dest.

ls lista de archivos en el directorio de trabajo.

ls dir lista de archivos en el directorio dir.

ls -a incluye archivos que empieza con un punto (.).

ls -l lista larga incluyendo permisos, tamaño y dueño.

ls -C lista en columnas.

ls -ld despliega información detallada en un directorio.

mail usuario envía un mail a usuario.

man command despliega las paginas del manual sobre command.

mkdir dir crea un directorio dir.

mv archivo1 archivo2 mueve el archivo1 a archivo2.

mv f1 f2 f3 dir mueve los archivos f1, f2 y f3 al directorio dir.

passwd cambia tu password.

pg archivo despliega archivo en pantalla por partes.

ps despliega estatus del proceso de la sesión concurrente.

ps -u usuario despliega procesos para usuario.

pwd muestra el directorio de trabajo en que te encuentras.

rcp host1:archivo host2:archivo copia archivos de un host a otro.

rm archivo borra archivo

rm -rdir borra el directorio dir y su contenido.

rmdir dir borra el directorio dir.

ruptime despliega el estatus de los hosts en la red.

sh bourne shell.

sort archivo desarrolla una clasificación alfabética del contenido de archivo.

sort -n archivo desarrolla una clasificación numérica del contenido de archivo.

sort +x archivo clasifica en el campo x + 1.

sort -ta archivo usa a como un campo separador.

spell archivo checa archivo para un correcto deletreo.

stty despliega las colocaciones stty concurrentes.

talk hablar con otro usuario desde tu terminal.

telnet host conecta a un host remoto.

tr a b archivo en archivo, cambia cada a por b.

vi archivo edita un archivo.

wc archivo lista el número de líneas, palabras y caracteres en archivo.

who despliega información de los usuario en la red.

write usuario envía un mensaje a la terminal del usuario.

II. REFERENCIA RÁPIDA COMANDOS vi

COMANDOS ESPECIALES

Esc regresa al modo de comando.

u deshace el último comando.

. repite el último insert, borrado o comando.

SALVAR TEXTO Y SALIR

:w escribe (salva) texto.

:w newfile guarda texto al archivo newfile.

: x,yw newfile guarda líneas de x hasta y en newfile.

:q! salir sin guardar cambios.

:wq guarda texto y sale.

POSICIÓN DEL CURSOR

N mueve a línea N

N+ baja N líneas.

N- sube N líneas.

^D baja una pantalla.

^U sube una pantalla.

k sube una línea.

j baja una línea.

^ inicio de línea.

\$ final del archivo.

Nw N palabras hacia adelante.

Nb regresa N palabras.

w una palabra hacia adelante.

b regresa una palabra.

e fin de palabra.

h regresa al espacio anterior.

l avanza un espacio.

flechitas espacio a la derecha o izquierda, arriba o hacia abajo una línea.

III. COMANDOS DE REFERENCIA TELNET

NOMBRE:

telnet - interface con el usuario del protocolo TELNET.

SINOPSIS:

telnet [nombre-de-la-máquina]

telnet itesocci

telnet eureka

DESCRIPCIÓN:

El comando telnet se comunica con otro host usando el protocolo TELNET. Si telnet es llamado sin argumentos, este entrará al modo de comando, indicado por su prompt (ejemplo, telnet >). En este modo, este acepta y ejecuta los comandos listados más adelante. Si este es invocado con argumentos, este representa un comando abierto (ver adelante) con estos argumentos.

Una vez que se ha establecido la conexión, telnet estará en el modo de entrada. En el modo de entrada, el texto escrito es inmediatamente enviado al host remoto para ser procesado.

En el modo de línea por línea, todo el texto es

COMANDOS TELNET

Los siguientes comandos son útiles.

telnet>

open host [port]

abre una conexión al host llamado. Si el número de puerto no es especificado, telnet intenta de conectar el servidor telnet desde el puerto default. La especificación del host puede ser tanto el nombre de un host o una dirección de internet especificada en "dot notation" (punto anotación)

close

cierra una sesión TELNET y te regresa al modo de comando.

quit

cierra cualquier sesión TELNET abierta y sale de telnet. Un fin de archivo (end-of-file) (en modo de comando) también cerrará una sesión y saldrá.

<Ctrl>z

suspende telnet. Este comando sólo trabaja cuando el usuario está usando csh o la el ambiente de aplicación BSD versión de ksh.

status

muestra el status actual de telnet.

display [argumento]

despliega todo, o algo, sobre

? [comand]

proporciona ayuda. Sin argumentos, telnet muestra un sumario de ayuda. Si un comando es especificado, telnet mostrará la información de ayuda sobre el comando.

send argumentos

envía uno o más secuencias de caracteres especiales a un host remoto. Los siguientes son argumentos los cuales pueden ser especificados (más de algún argumento puede ser especificado en un tiempo).

escape

envía el caracter telnet escape.

synch

envía la secuencia SYNCH TELNET. Esta secuencia causa que el sistema remoto descarte todo lo previamente tecleado como entrada, pero que todavía no haya sido leído. Esta secuencia es enviada como un dato urgente TCP.

brk

envía la secuencia TELNET BRK (break - rompimiento), la cual puede tener significancia para el sistema remoto.

ip

envía la secuencia TELNET IP (interrupción de proceso), la cual debe causar que el sistema remoto aborte en proceso que se esta corriendo.

ao

envía la secuencia TELNET AO (abortar salida), la cual puede causar al sistema remoto que nivele todas las salidas del sistema remoto a la terminal del usuario.

ayt

envía la secuencia TELNET AYT (are you there- estas ahí), el cual el sistema moto puedo o no puede responder.

ec

envía la secuencia TELNET EC (erase character- borrar caracter), la cual puede causar al sistema remoto a borrar el último caracter tecleado.

el

envía la secuencia TELNET EL (erase line - borrar línea), la cual causa que el sistema remoto borre la línea anterior escrita.

ga

envía la secuencia TELNET GA (go ahead - adelante), la cual probablemente no tiene significado para el sistema remoto.

nop

envía la secuencia TELNET NOP (no operación - no operación).

?

imprime información de ayuda sobre el comando enviado.